FIG. 1A

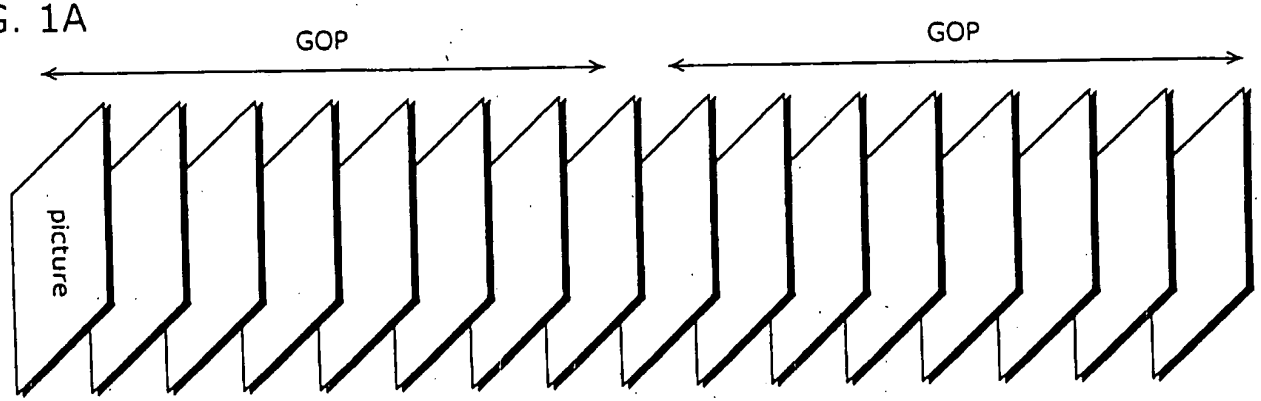

GOP
GOP
picture

FIG. 1B

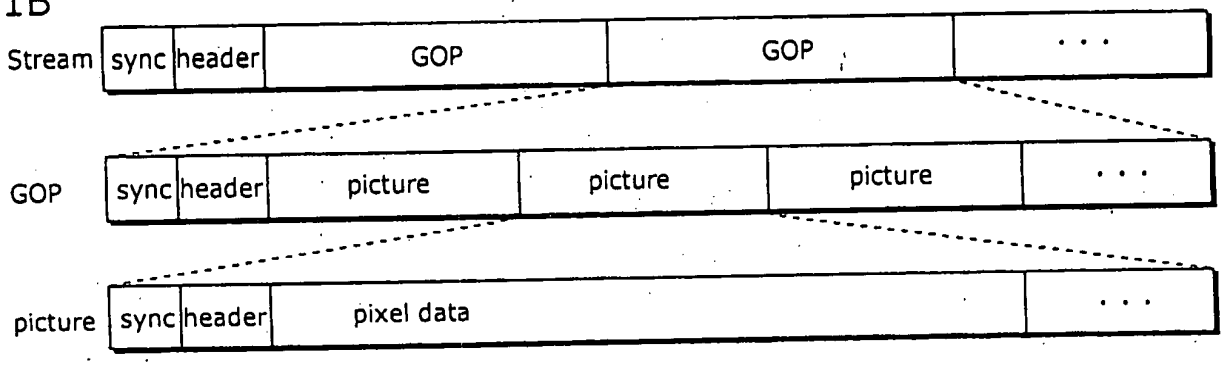

Stream
sync
header
GOP
GOP
GOP
sync
header
picture
picture
picture
picture
sync
header
pixel data

FIG. 2A

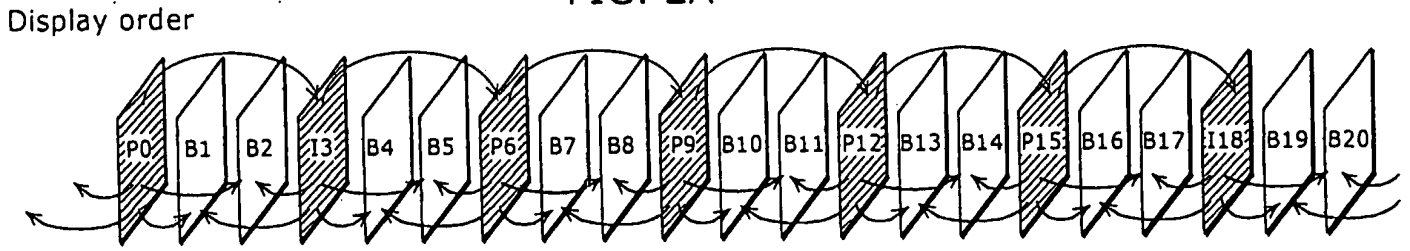

Display order
P0
B1
B2
I3
B4
B5
P6
B7
B8
P9
B10
B11
P12
B13
B14
P15
B16
B17
I18
B19
B20

FIG. 2B

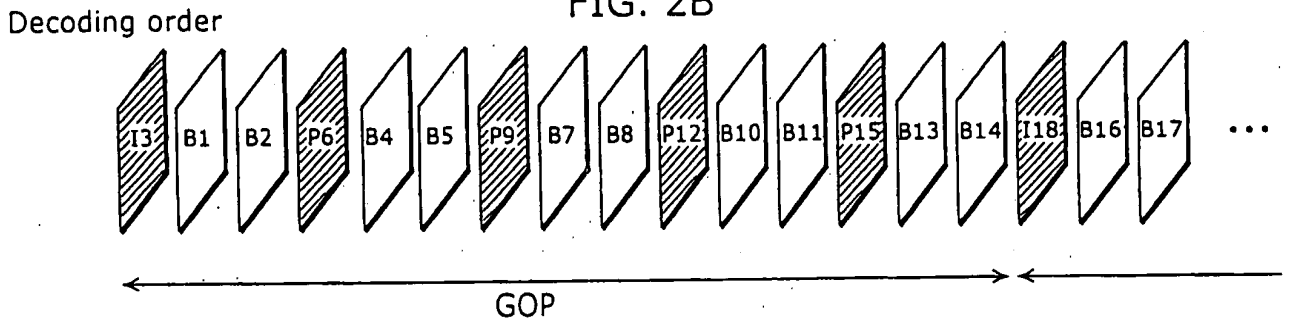

Decoding order
I3
B1
B2
P6
B4
B5
P9
B7
B8
P12
B10
B11
P15
B13
B14
I18
B16
B17
...
GOP

FIG. 3A

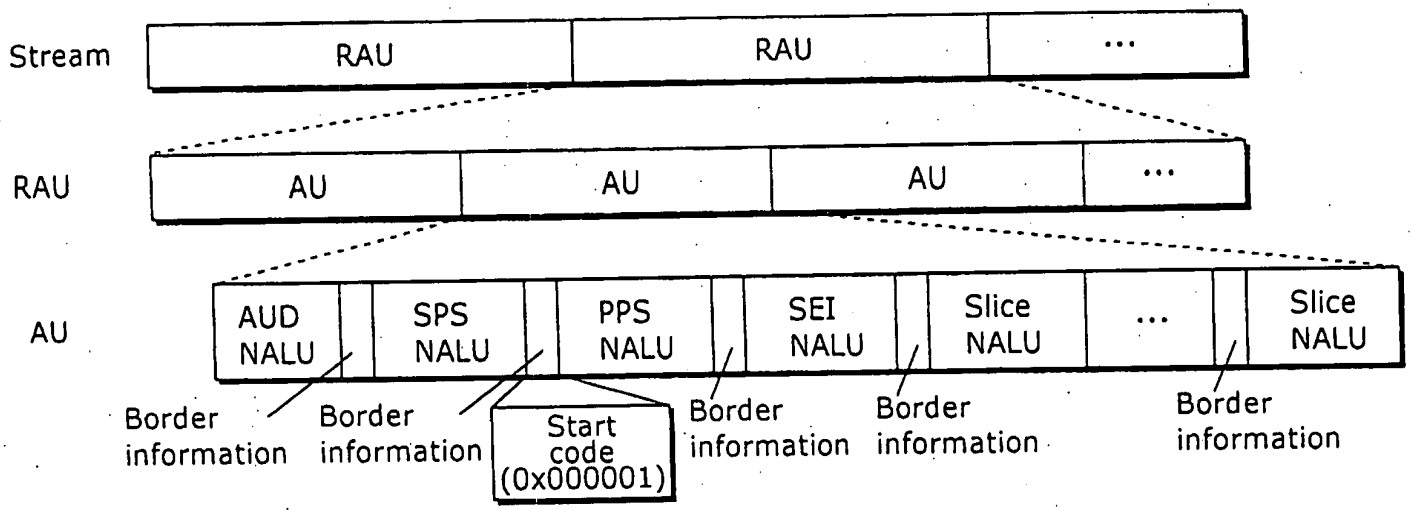

Stream
RAU
RAU
...
RAU
AU
AU
AU
...
AU
AUD NALU
SPS NALU
PPS NALU
SEI NALU
Slice NALU
...
Slice NALU
Border information
Border information
Start code (0x000001)
Border information
Border information
Border information

FIG. 3B

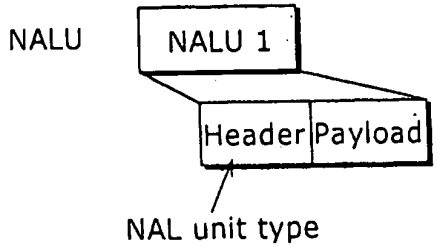

NALU
NALU 1
Header
Payload
NAL unit type

FIG. 4

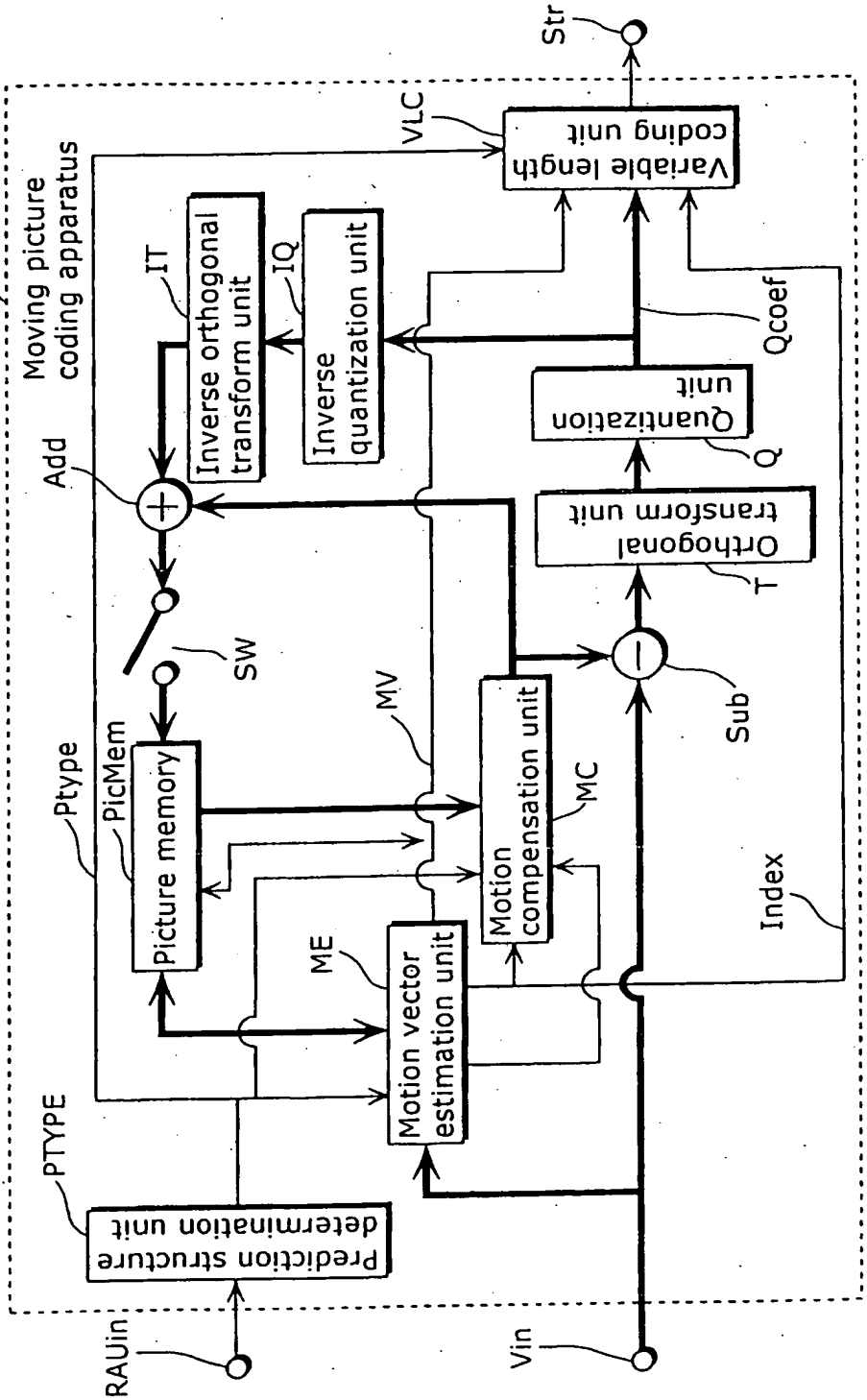

1
Moving picture coding apparatus
Str
VLC
Variable length coding unit
IT
Inverse orthogonal transform unit
IQ
Inverse quantization unit
Qcoef
Quantization unit
Q
Add
Orthogonal transform unit
T
SW
Sub
MV
Ptype
PicMem
Picture memory
Motion compensation unit
MC
ME
Motion vector estimation unit
Index
PTYPE
Prediction structure determination unit
RAUin
Vin

FIG. 5

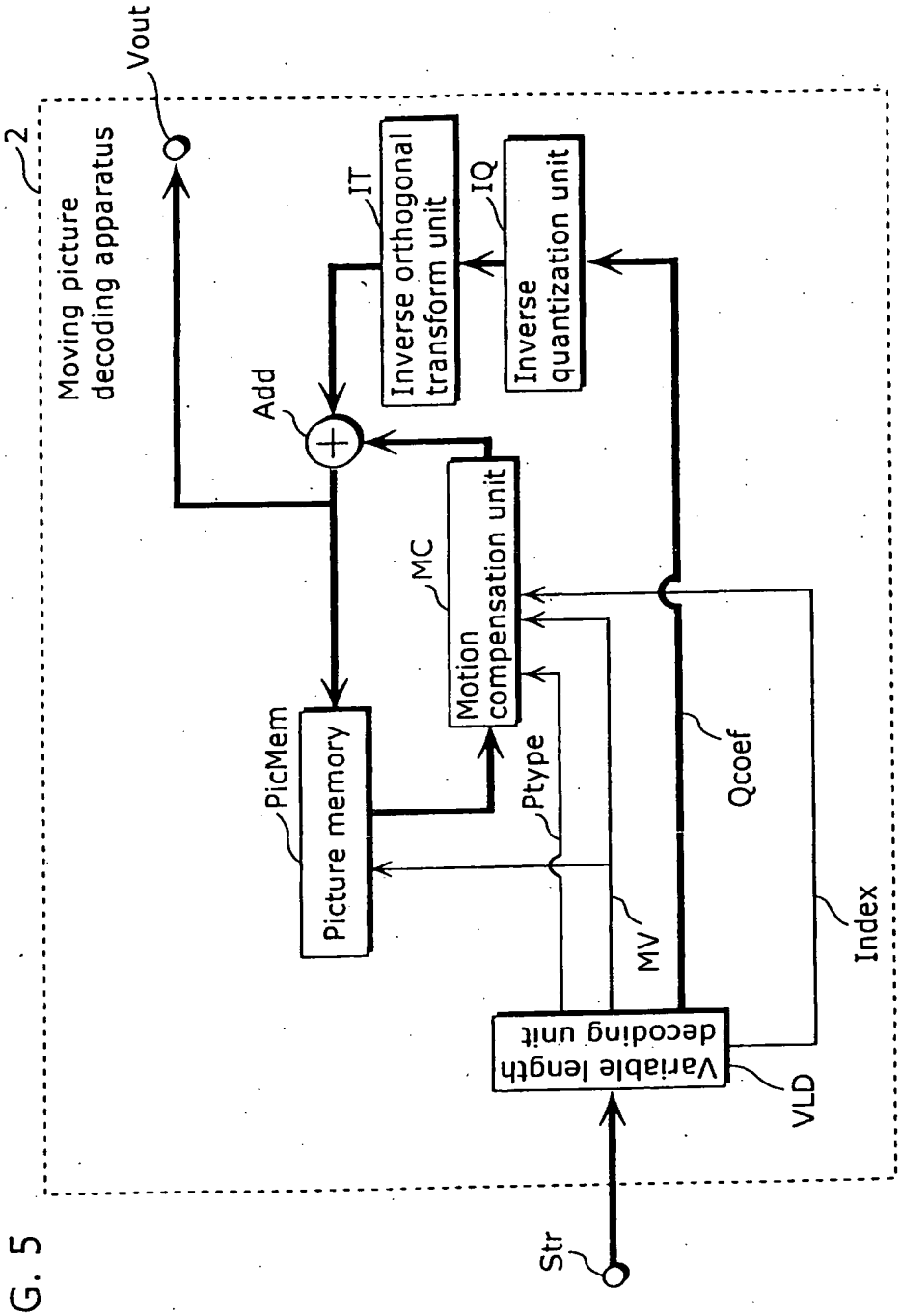

2
Moving picture decoding apparatus
Vout
IT
Inverse orthogonal transform unit
IQ
Inverse quantization unit
Add
MC
Motion compensation unit
PicMem
Picture memory
Ptype
Qcoef
MV
Index
Variable length decoding unit
VLD
Str

FIG. 7

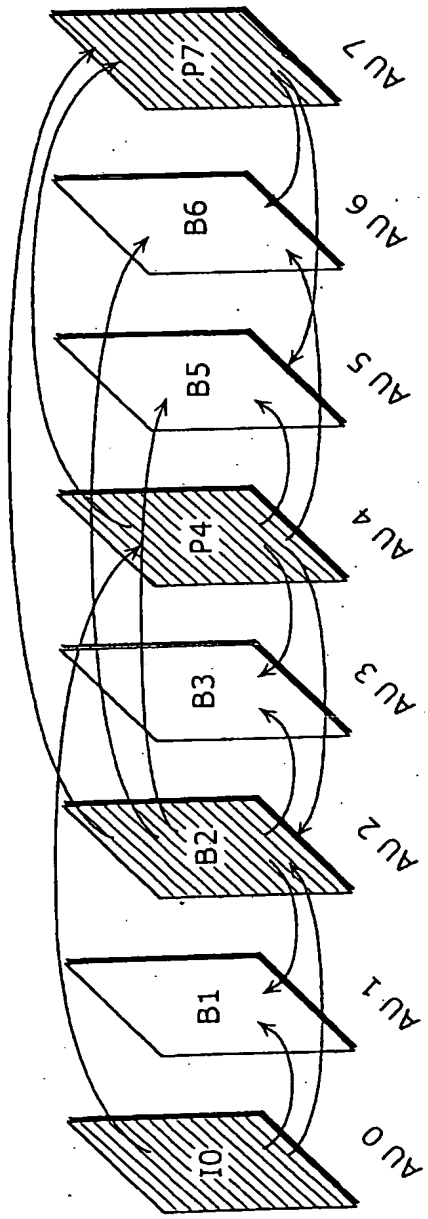


FIG. 8A

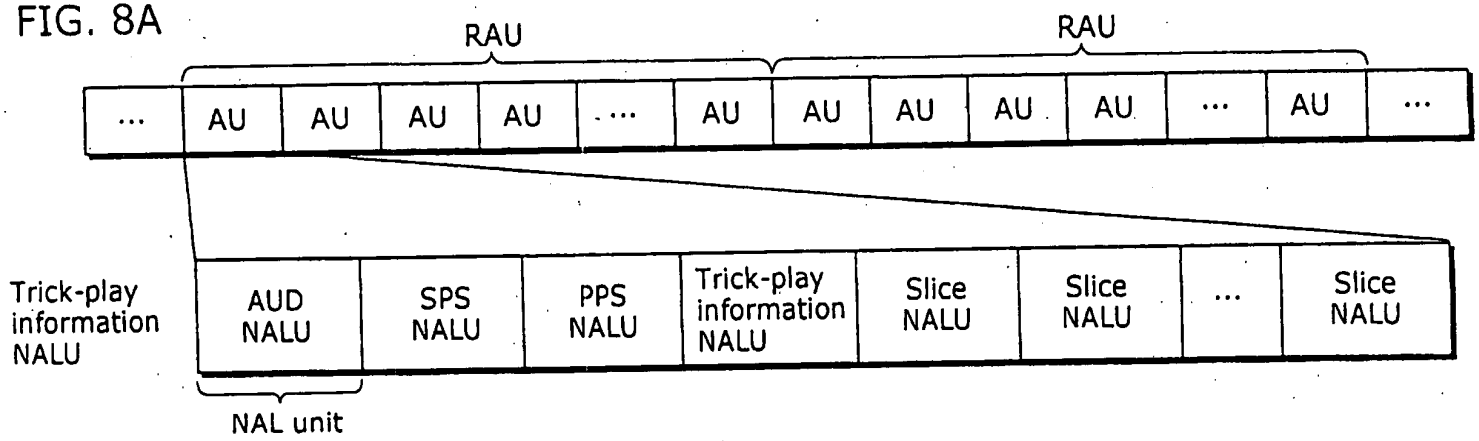

RAU
RAU
...
AU
AU
AU
AU
AU
AU
AU
AU
AU
AU
...
AU
...
Trick-play information NALU
AUD NALU
SPS NALU
PPS NALU
Trick-play information NALU
Slice NALU
Slice NALU
...
Slice NALU
NAL unit

FIG. 8B

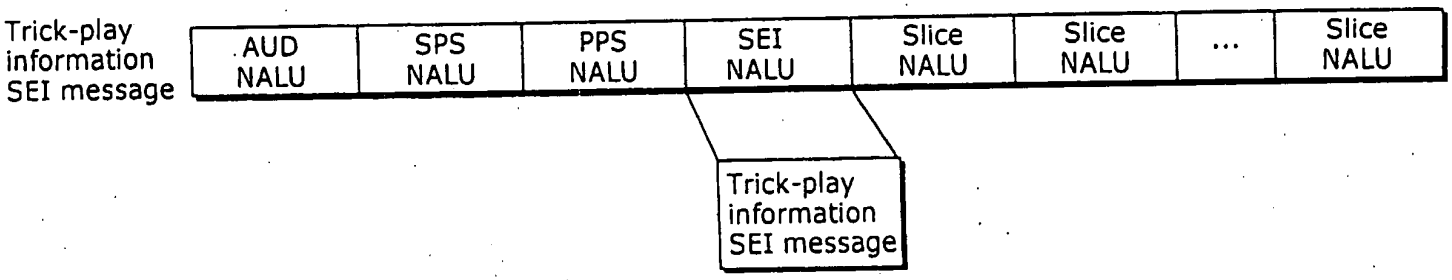

Trick-play information SEI message
AUD NALU
SPS NALU
PPS NALU
SEI NALU
Slice NALU
Slice NALU
...
Slice NALU
Trick-play information SEI message

FIG. 9A

Display order

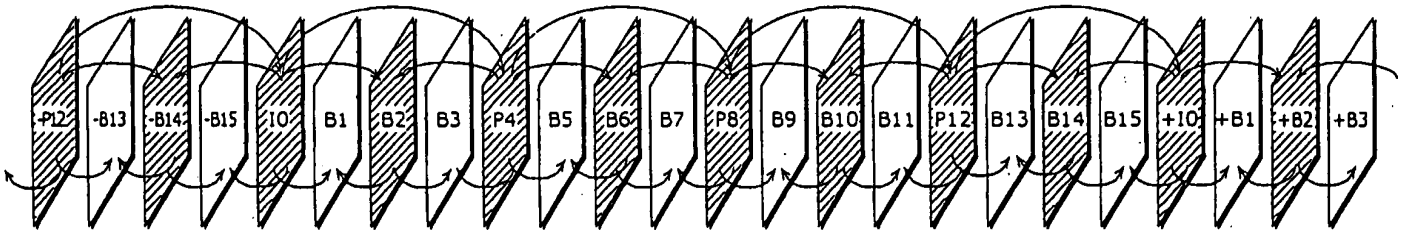


FIG. 9B

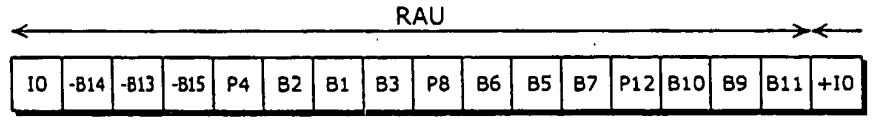


FIG. 9C

FIG. 9D

FIG. 10A

Display order

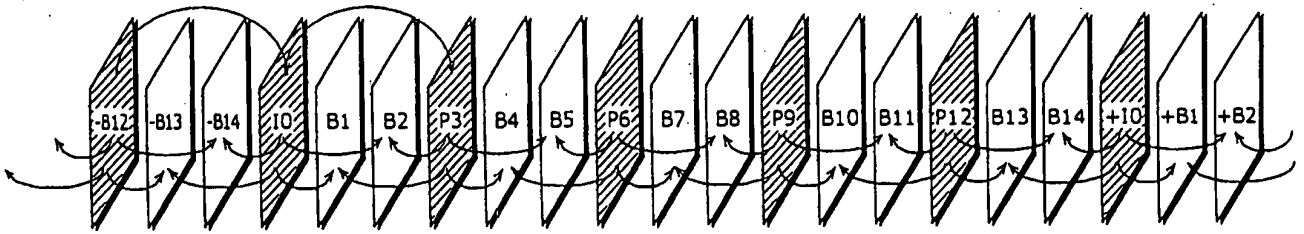


FIG. 10B

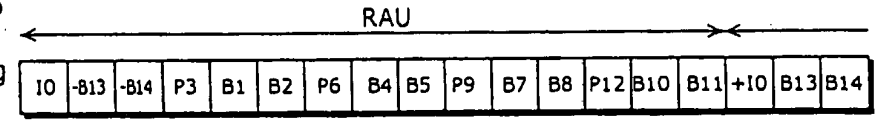


FIG. 10C

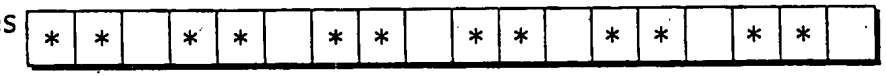


FIG. 10D

FIG. 11A

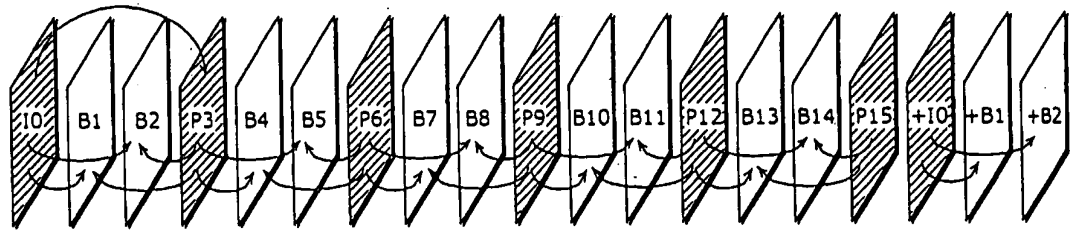


FIG. 11B

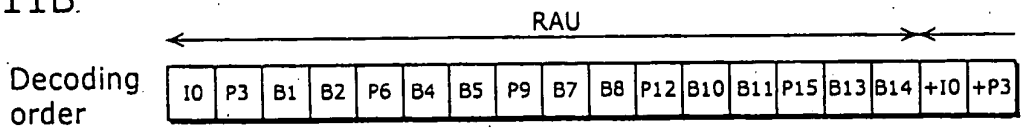


FIG. 11C

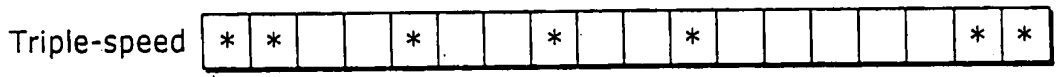


FIG. 12A

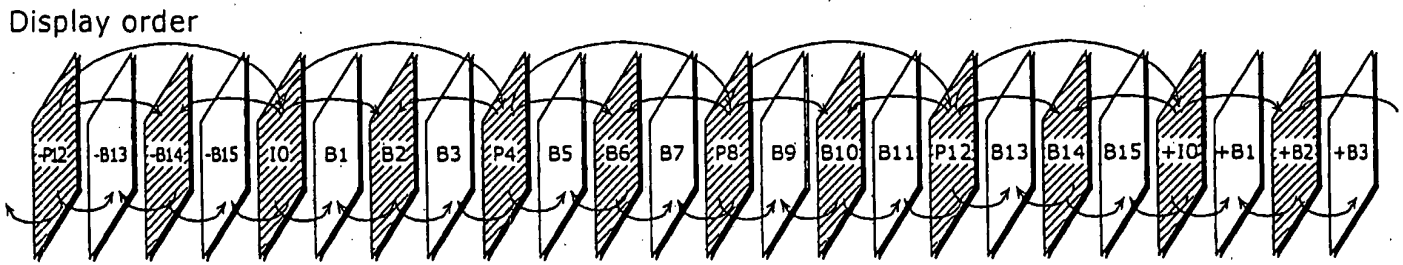


FIG. 12B

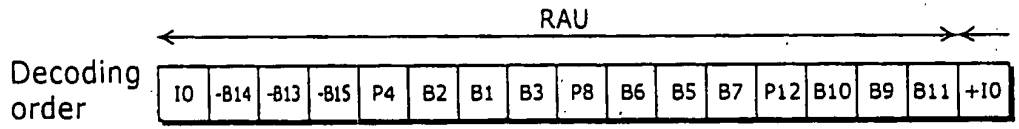


FIG. 12C

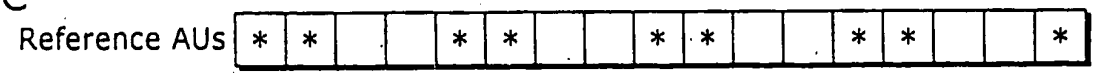


FIG. 12D

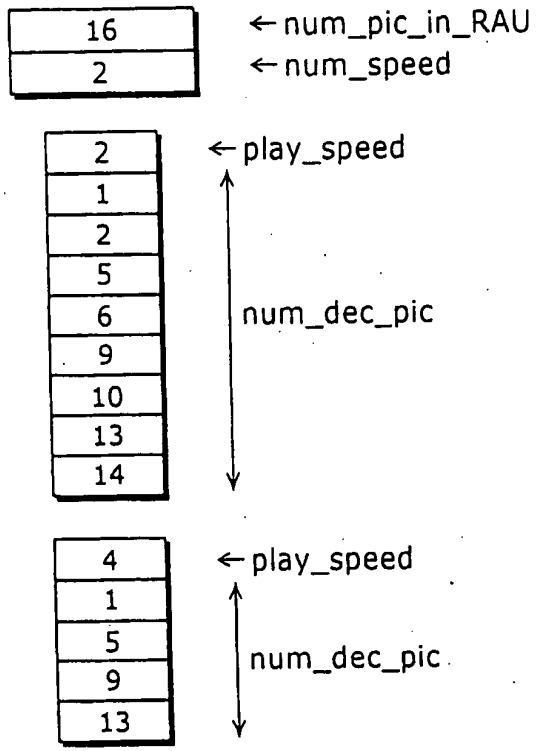
FIG. 12E

FIG. 12F

```
Variable Speed Play {
  num_pic_in_RAU;
  num_speed;
  for (i=0; i < num_speed; i++) {
    play_speed;
    num_dec_pic;
    for (j=0; j < num_dec_pic; j++) {
      dec_pic;
    }
  }
}
```

Syntax example

FIG. 13A



Data storage unit

FIG. 13B

FIG. 14

```
Variable Speed Play {
  num_pic_in_RAU;
  num_speed;
  for (i=0; i < num_speed; i++) {
    play_speed;
    num_dec_pic;
    pts_dts_flag;
    for (j=0; j < num_dec_pic; j++) {
      dec_pic;
      if (pts_dts_flag) diplay_order;
    }
  }
}
```

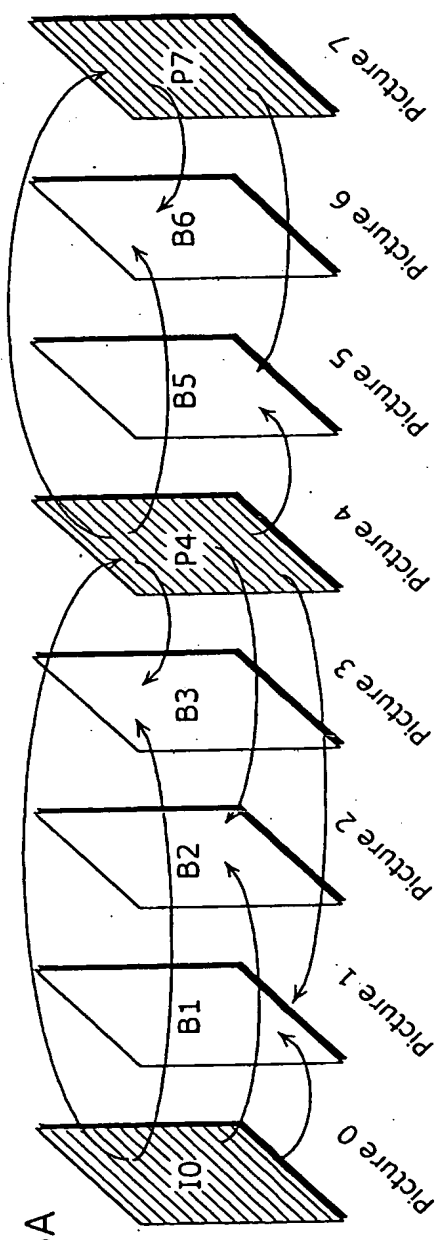

I0
B1
B2
B3
P4
B5
B6
P7
picture 0
picture 1
picture 2
picture 3
picture 4
picture 5
picture 6
picture 7

FIG. 6A

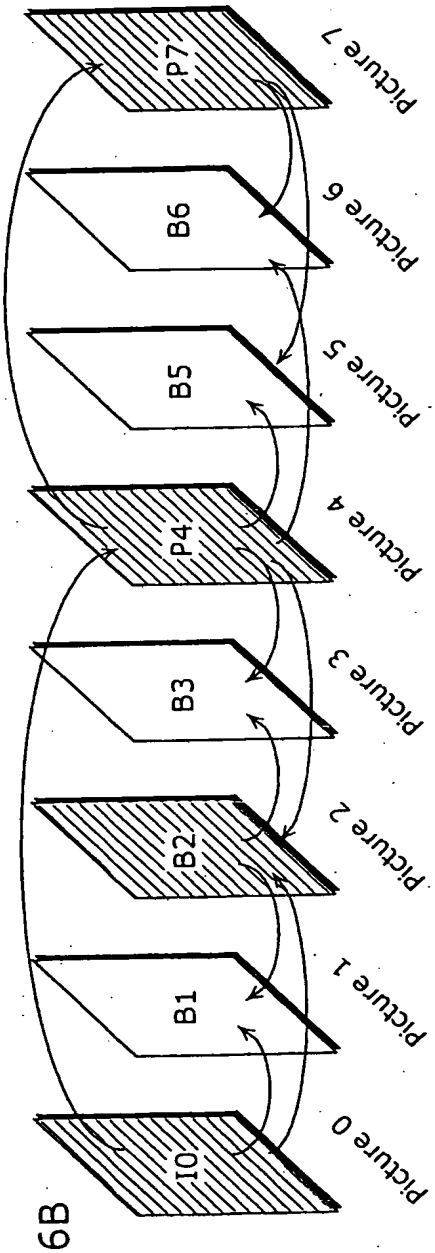

I0
B1
B2
B3
P4
B5
B6
P7
picture 0
picture 1
picture 2
picture 3
picture 4
picture 5
picture 6
picture 7

FIG. 6B

FIG. 15A

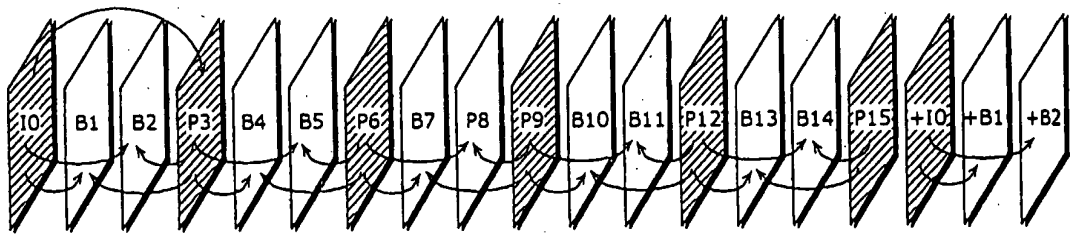


FIG. 15B

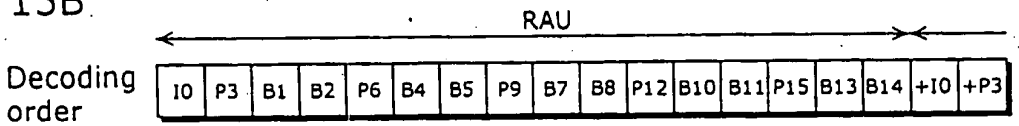


FIG. 15C

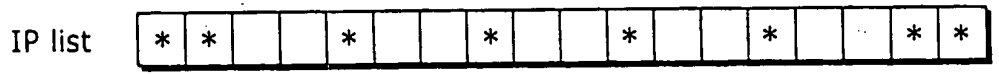


FIG. 16A

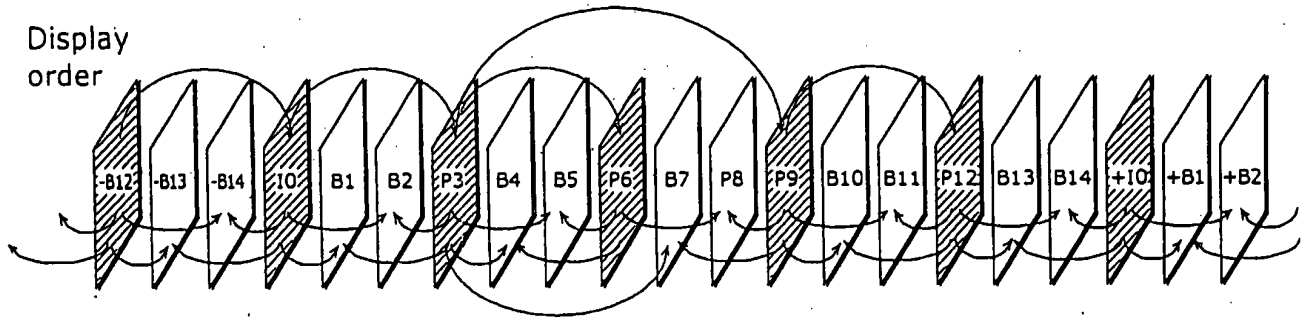


FIG. 16B

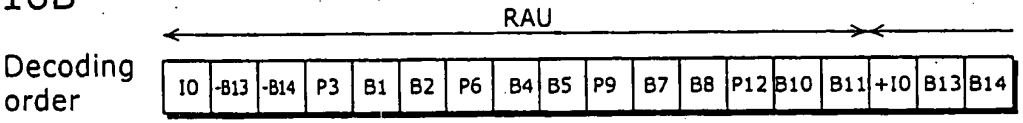


FIG. 16C

FIG. 17A

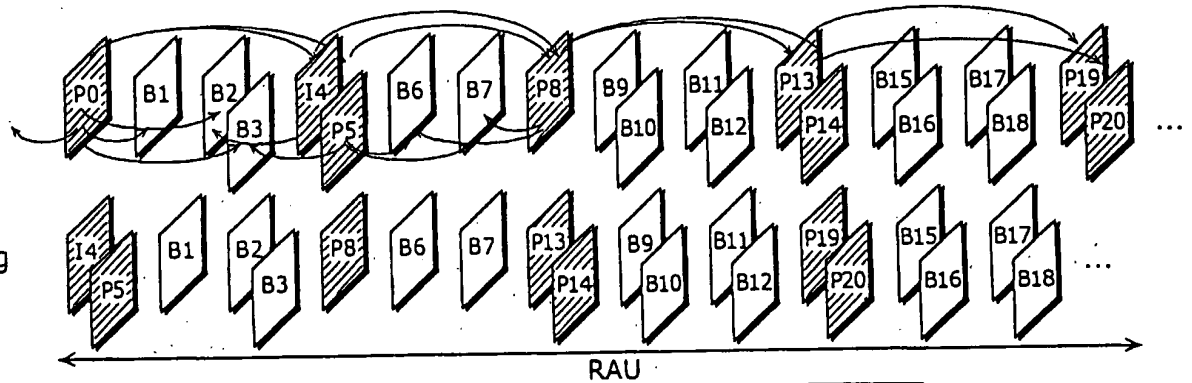


FIG. 17B

FIG. 17C

```
RAU map {
  num_AU_in_RAU;
  for (i=0; i < num_AU_in_RAU; i++) {
      frame_field_flag;
      pic_type;
  }
}
```

FIG. 17F

```
RAU map {
  num_frame_in_RAU;
  for (i=0; i < num_frame_in_RAU; i++) {
      frame_flag;
      if (frame_field_flag) frame_type;
      else field_pair_type;
  }
}
```

FIG. 17D

30 ←num_AU_in_RAU

frame_field_flag	pic_type	
0	0	←I4
0	1	←P5
1	3	←B1
0	3	←B2
0	3	←B3
1	1	←P8
1	3	←B6
1	3	←B7
0	1	←P13
0	1	←P14
0	3	←B9
0	3	←B10
0	3	←B11
0	3	←B12
0	1	←P19
0	1	←P20
...	...	

15 ←num_frame_in_RAU

frame_flag	frame_type	field_pair_type	
0	-	IP	←I4,P5
1	3	-	←B1
0	-	BnBn	←B2,B3
1	1	-	←P8
1	3	-	←B6
1	3	-	←B7
0	-	PP	←P13,P14
0	-	BnBn	←B9,B10
0	-	BnBn	←B11,B12
0	-	PP	←P19,P20
0	3	BnBn	←B15,B16
0	3	BnBn	←B17,B18
...	...	...	

FIG. 17E

FIG. 18A

```
RAU map {
        num_AU_in_RAU;
        for (i=0; i < num_AU_in_RAU; i++) {
                picture_structure;
                picture_type;
        }
}
```

FIG. 18B

```
picture_structure:    Field
                   or Frame
                      .
                      .
                      .
```

FIG. 18C

```
picture_type:      I picture
                   or Reference B picture
                   or Non-reference B picture
                   or P picture
                      .
                      .
                      .
```

FIG. 19

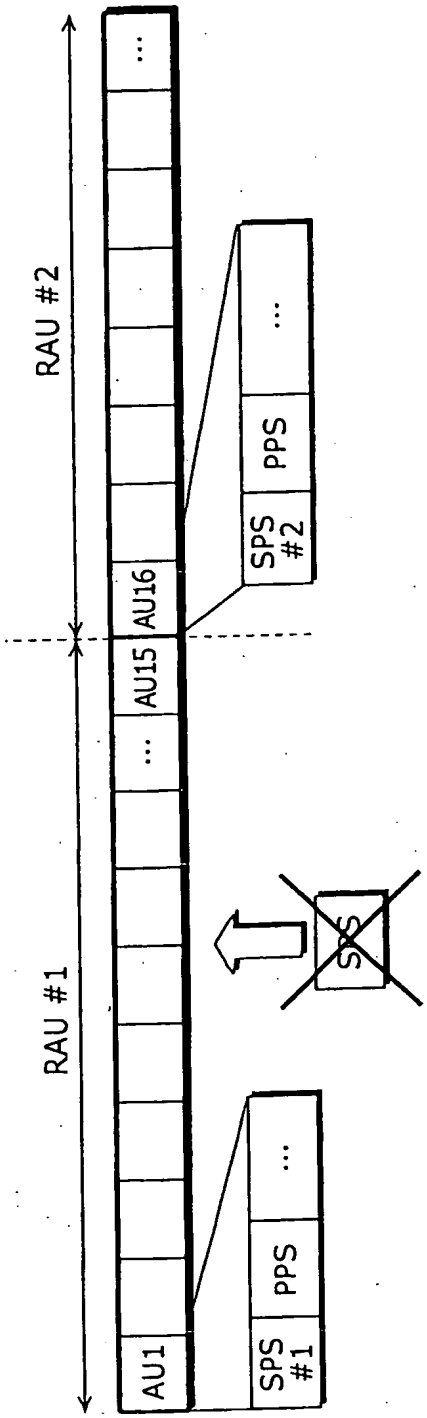


FIG. 20A

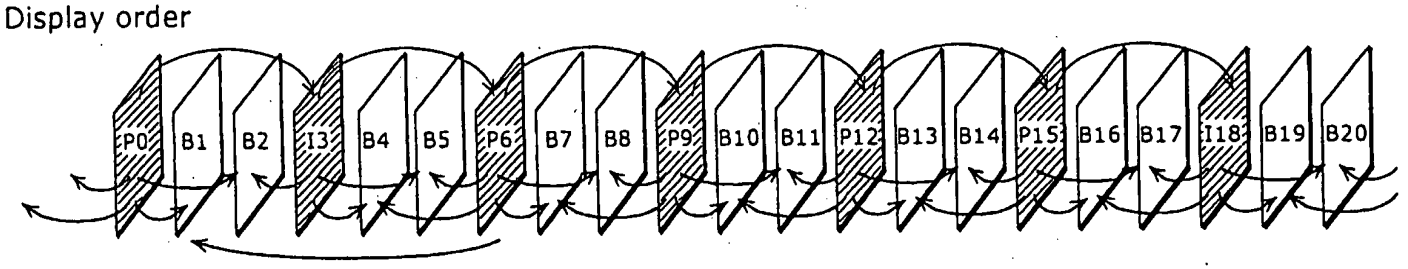
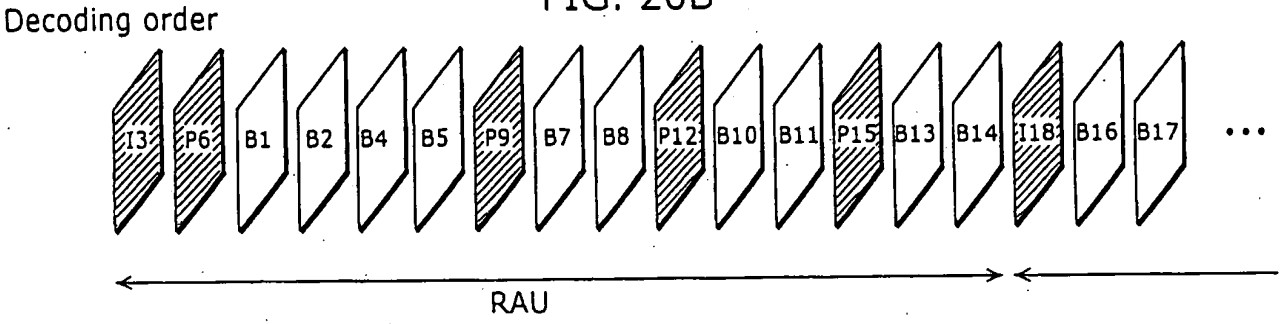


FIG. 20B



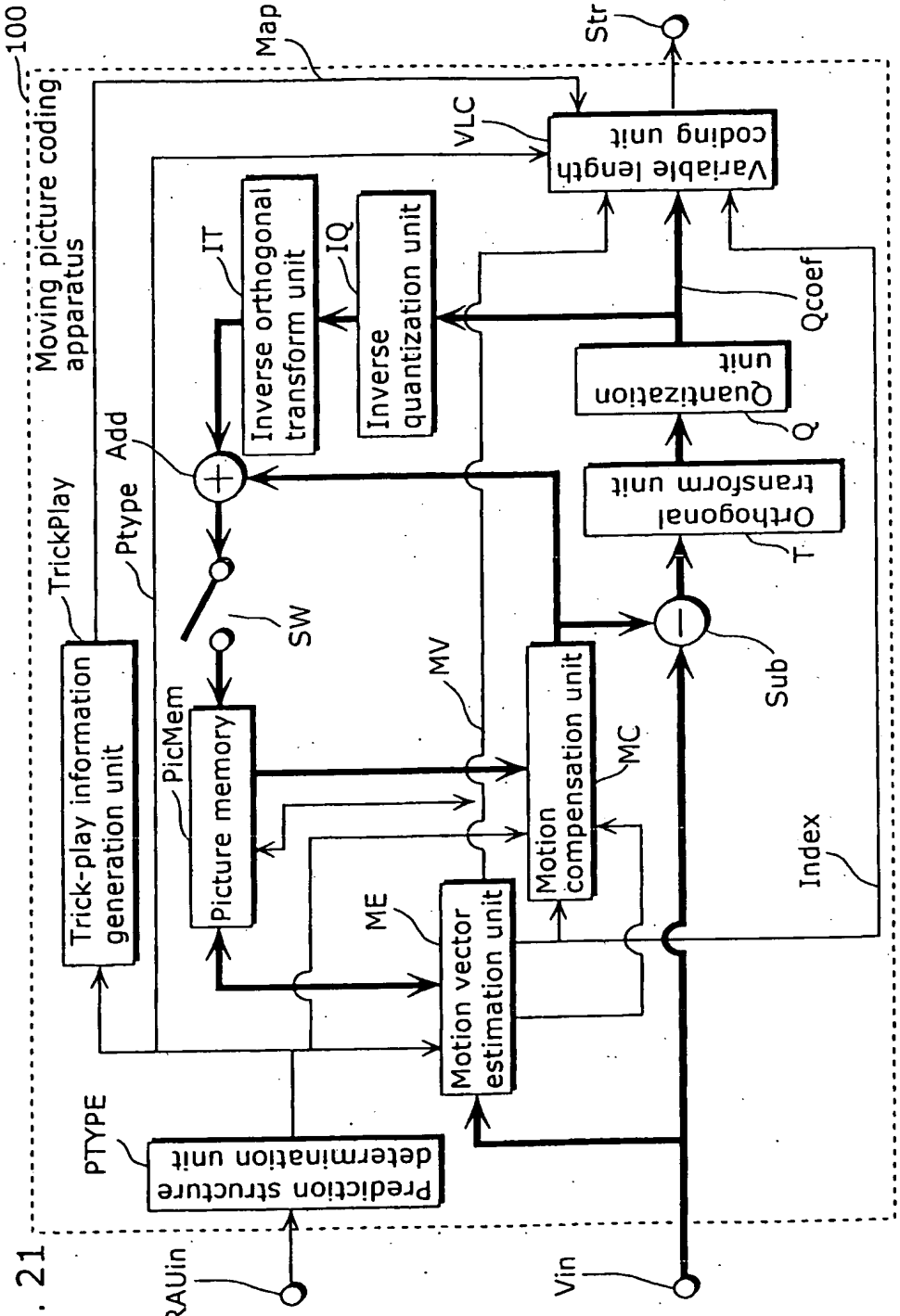

100
Moving picture coding apparatus
Map
Str
VLC
Variable length coding unit
IT
Inverse orthogonal transform unit
IQ
Inverse quantization unit
Qcoef
Quantization unit
Q
Add
Orthogonal transform unit
T
TrickPlay
Ptype
SW
Sub
Trick-play information generation unit
PicMem
Picture memory
MV
Motion compensation unit
MC
ME
Motion vector estimation unit
Index
PTYPE
Prediction structure determination unit
RAUin
Vin

FIG. 21

FIG. 22

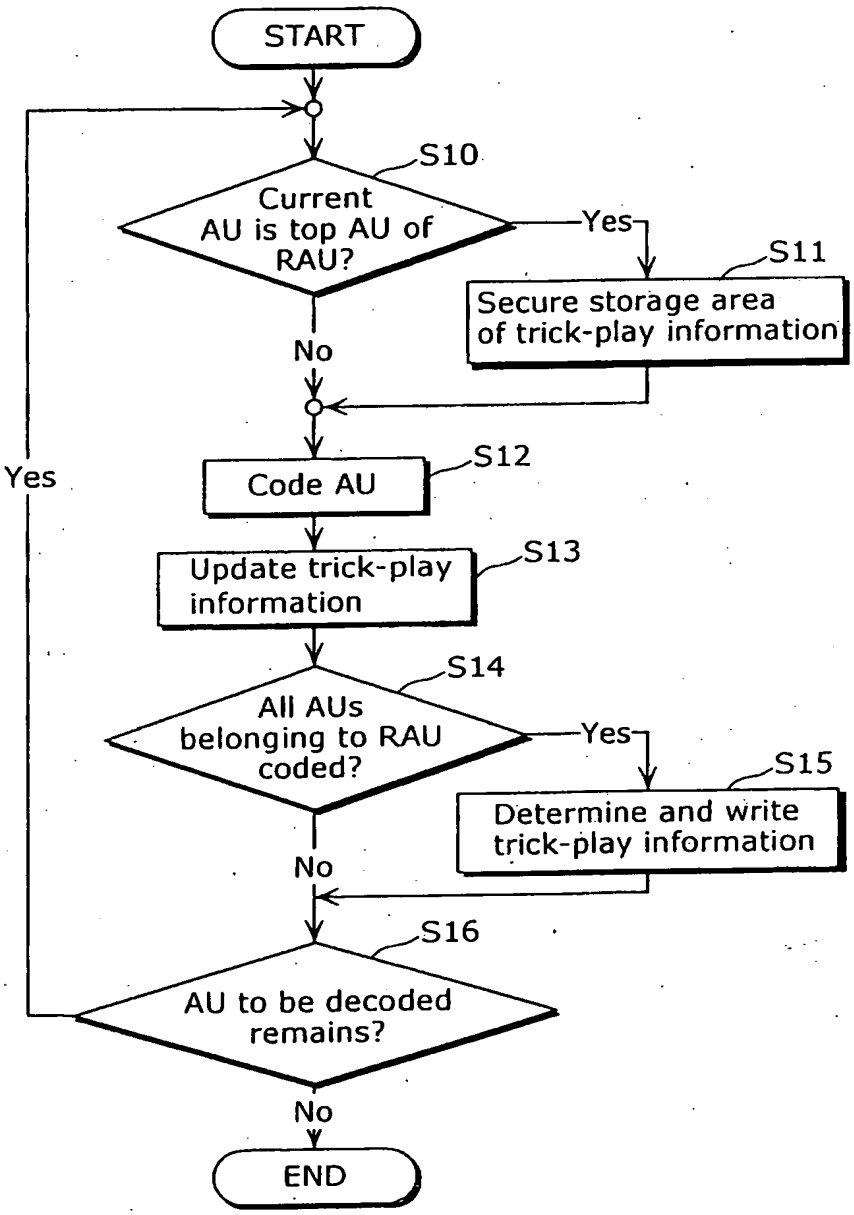

START
S10
Current AU is top AU of RAU?
Yes
S11
Secure storage area of trick-play information
No
Code AU
S12
Update trick-play information
S13
All AUs belonging to RAU coded?
S14
Yes
S15
Determine and write trick-play information
No
AU to be decoded remains?
S16
Yes
No
END

FIG. 23

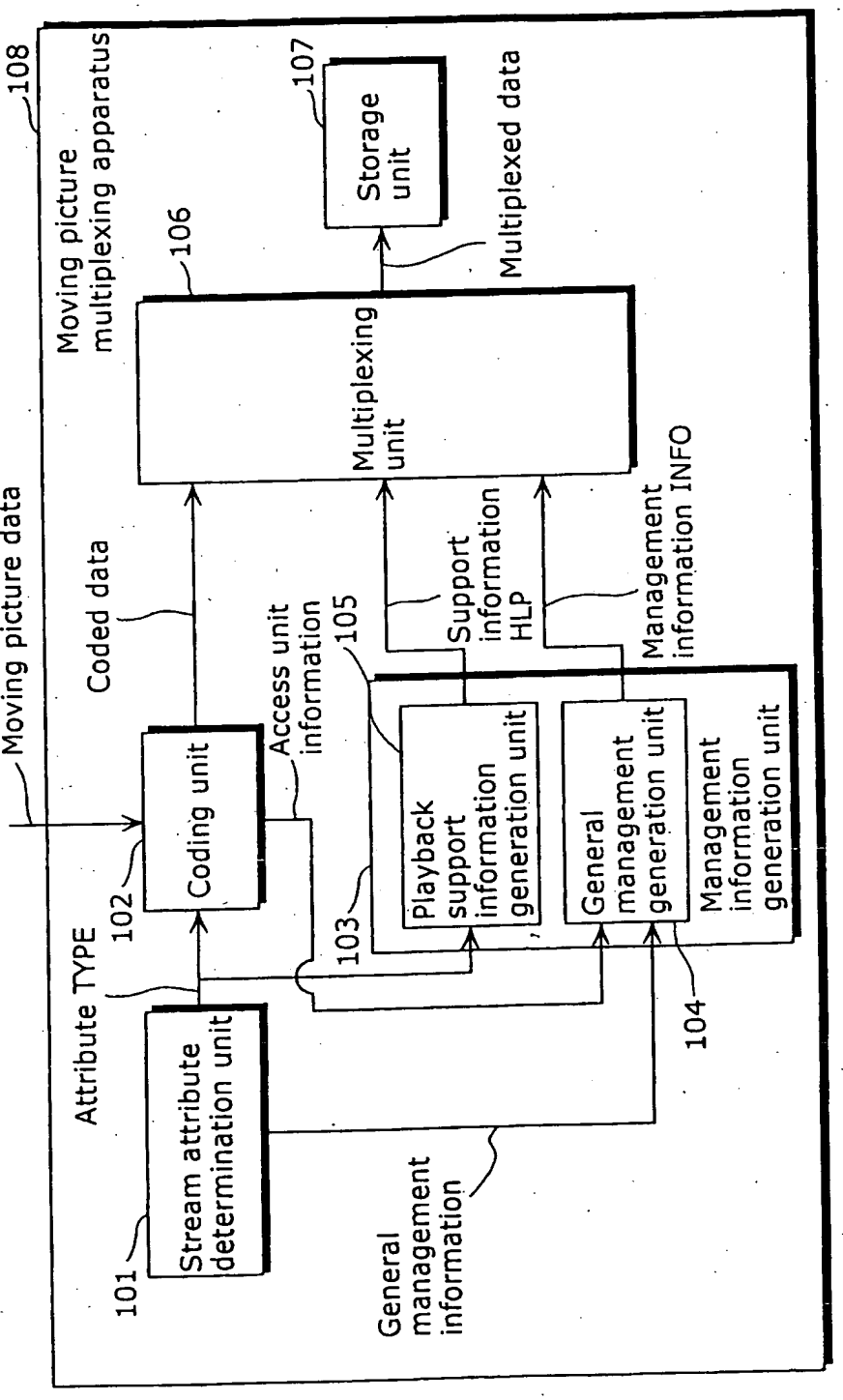

108
Moving picture multiplexing apparatus
Moving picture data
Attribute TYPE
102
Coding unit
Coded data
Access unit information
105
Support information HLP
Management information INFO
106
Multiplexing unit
Multiplexed data
107
Storage unit
103
Playback support information generation unit
General management generation unit
Management information generation unit
104
101
Stream attribute determination unit
General management information

FIG. 24A

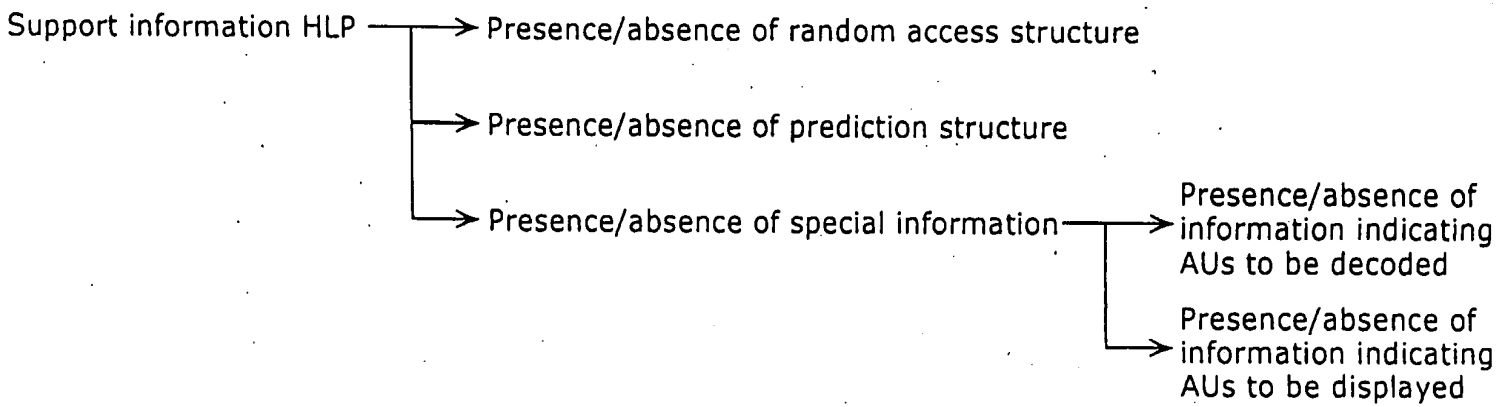


FIG. 24B

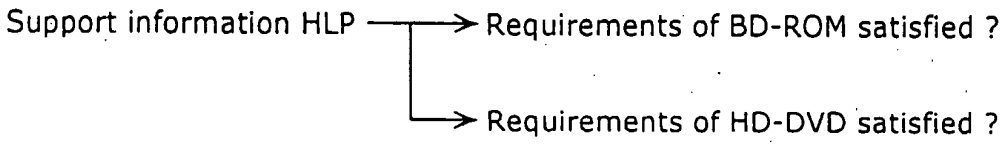


FIG. 25

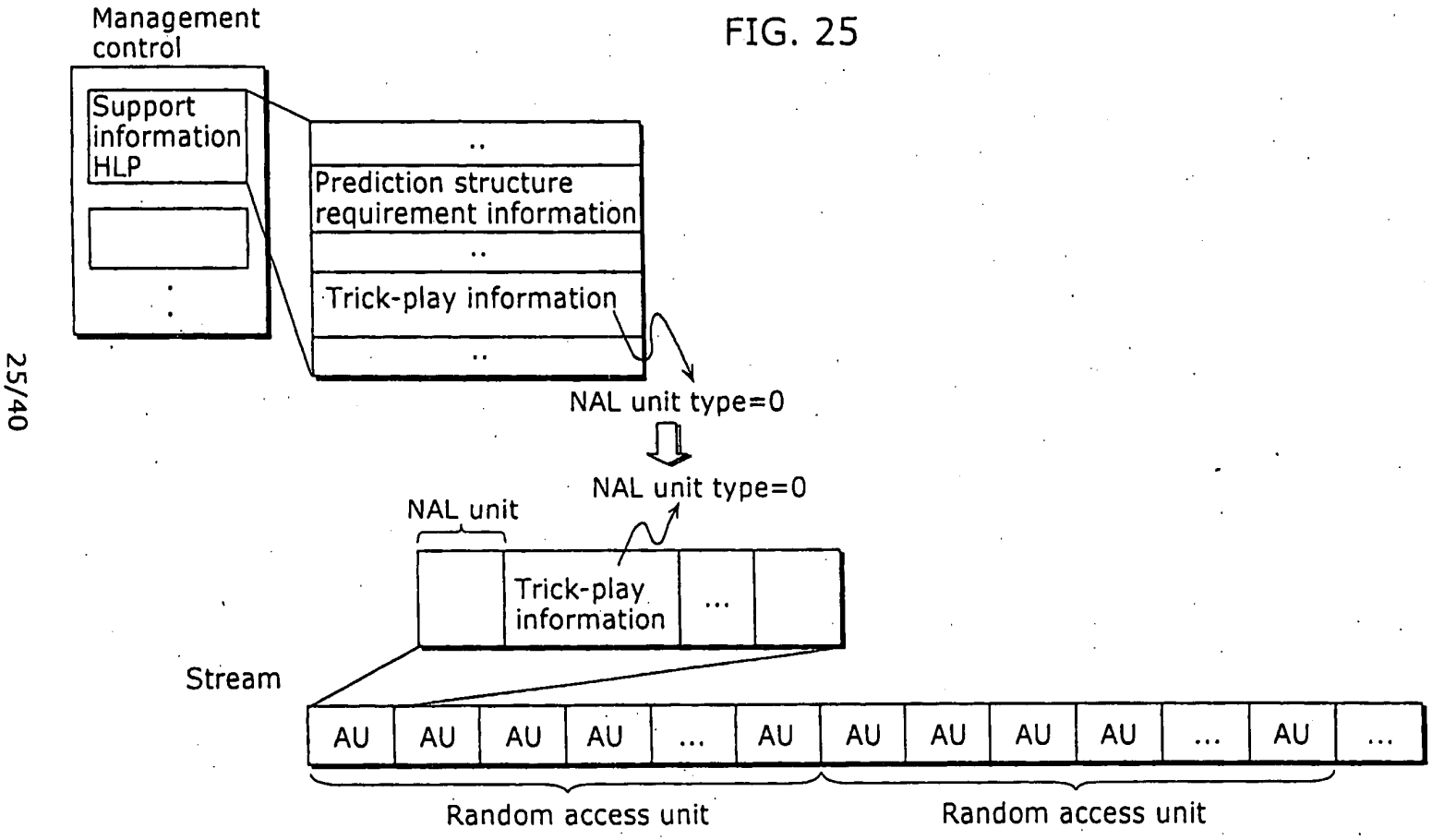

Management control
Support information HLP
Prediction structure requirement information
Trick-play information
NAL unit type=0
NAL unit type=0
NAL unit
Trick-play information
Stream
AU
Random access unit
Random access unit

FIG. 26

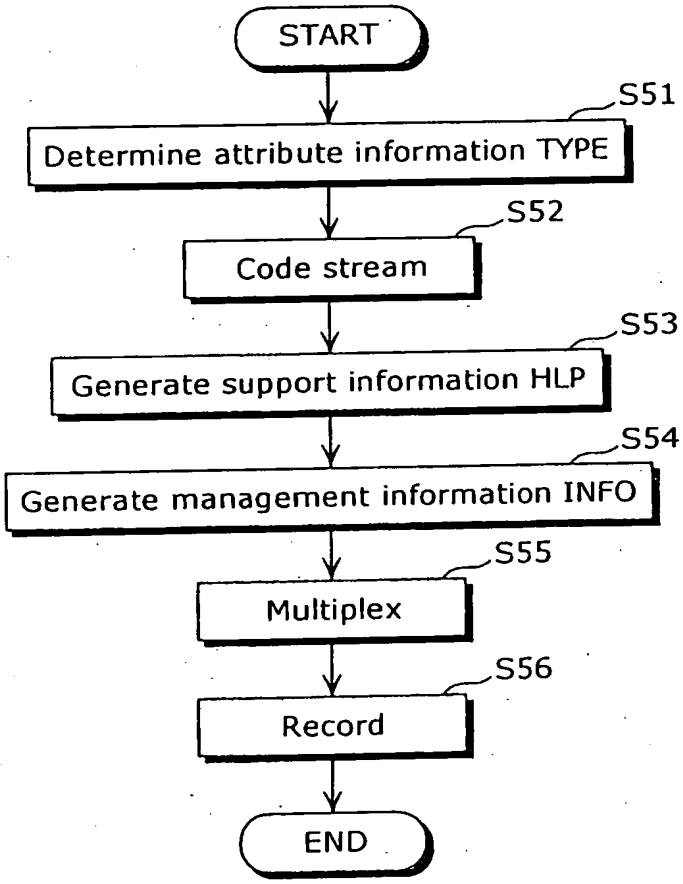

START
S51
Determine attribute information TYPE
S52
Code stream
S53
Generate support information HLP
S54
Generate management information INFO
S55
Multiplex
S56
Record
END

FIG. 27

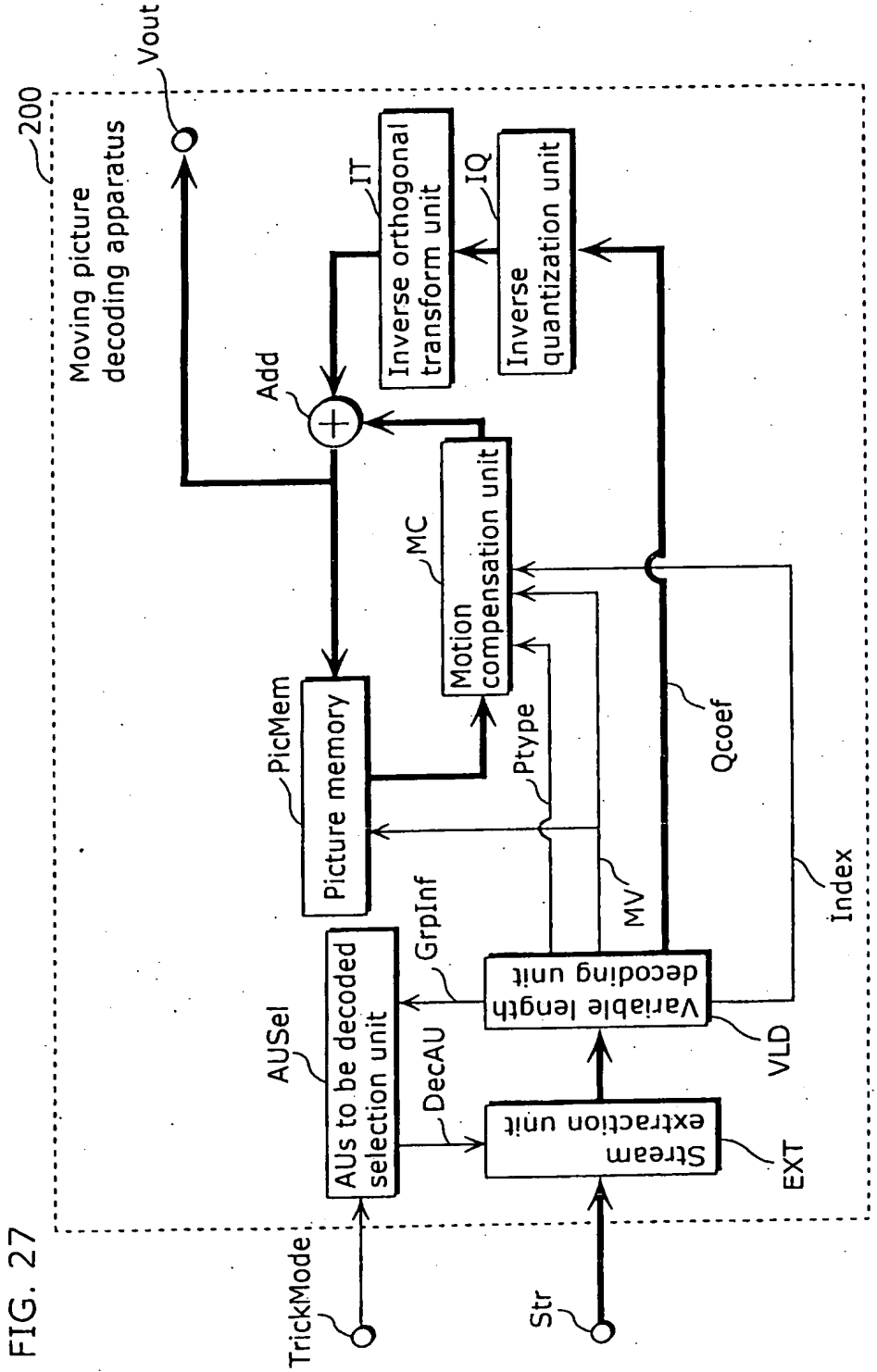

200
Moving picture decoding apparatus
Vout
IT
Inverse orthogonal transform unit
IQ
Inverse quantization unit
Add
MC
Motion compensation unit
PicMem
Picture memory
Ptype
Qcoef
MV
Index
GrpInf
AUSel
AUs to be decoded selection unit
DecAU
Variable length decoding unit
VLD
Stream extraction unit
EXT
TrickMode
Str

FIG. 28

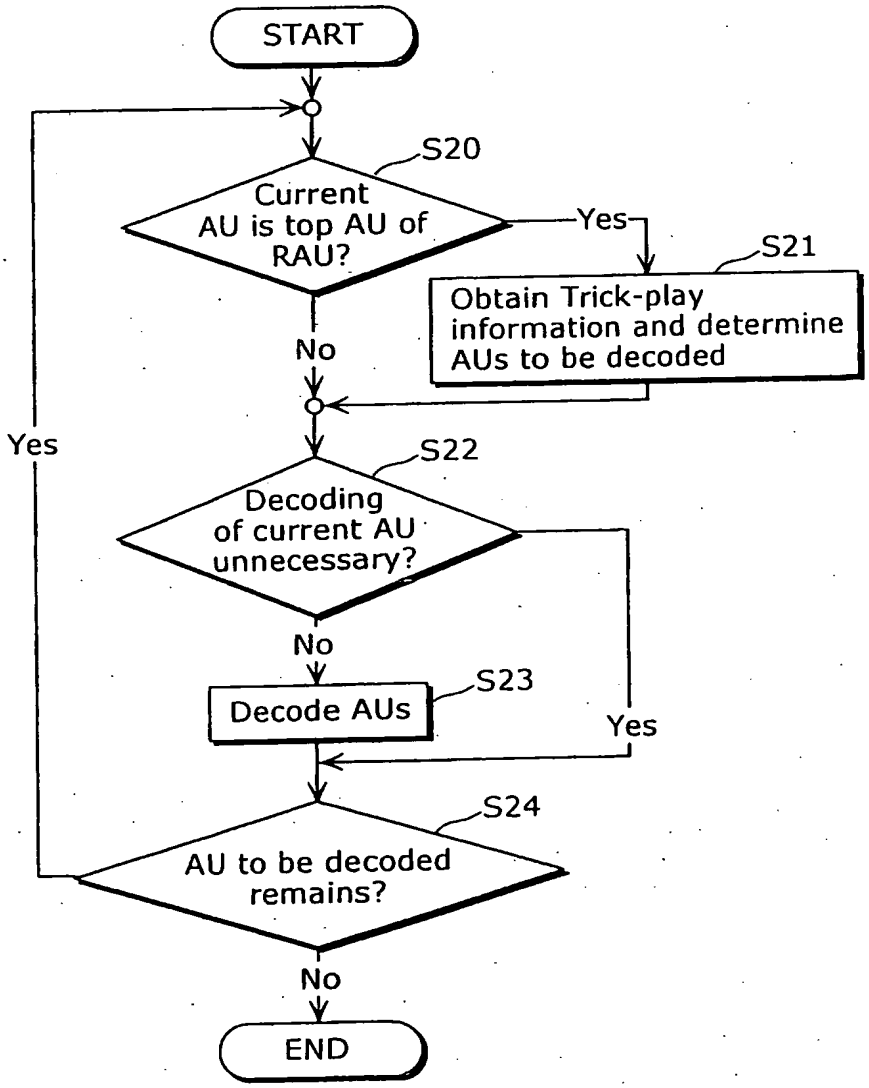

START
S20
Current AU is top AU of RAU?
Yes
S21
Obtain Trick-play information and determine AUs to be decoded
No
S22
Decoding of current AU unnecessary?
Yes
No
S23
Decode AUs
Yes
S24
AU to be decoded remains?
Yes
No
END

FIG. 29

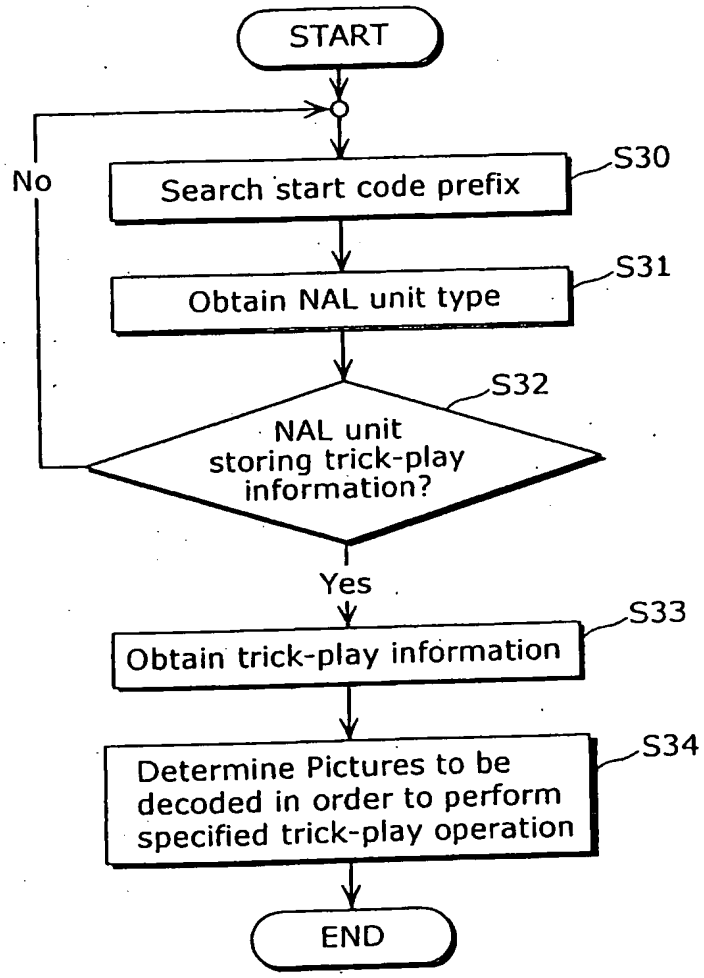

START
Search start code prefix
S30
Obtain NAL unit type
S31
NAL unit storing trick-play information?
S32
No
Yes
Obtain trick-play information
S33
Determine Pictures to be decoded in order to perform specified trick-play operation
S34
END

FIG. 30

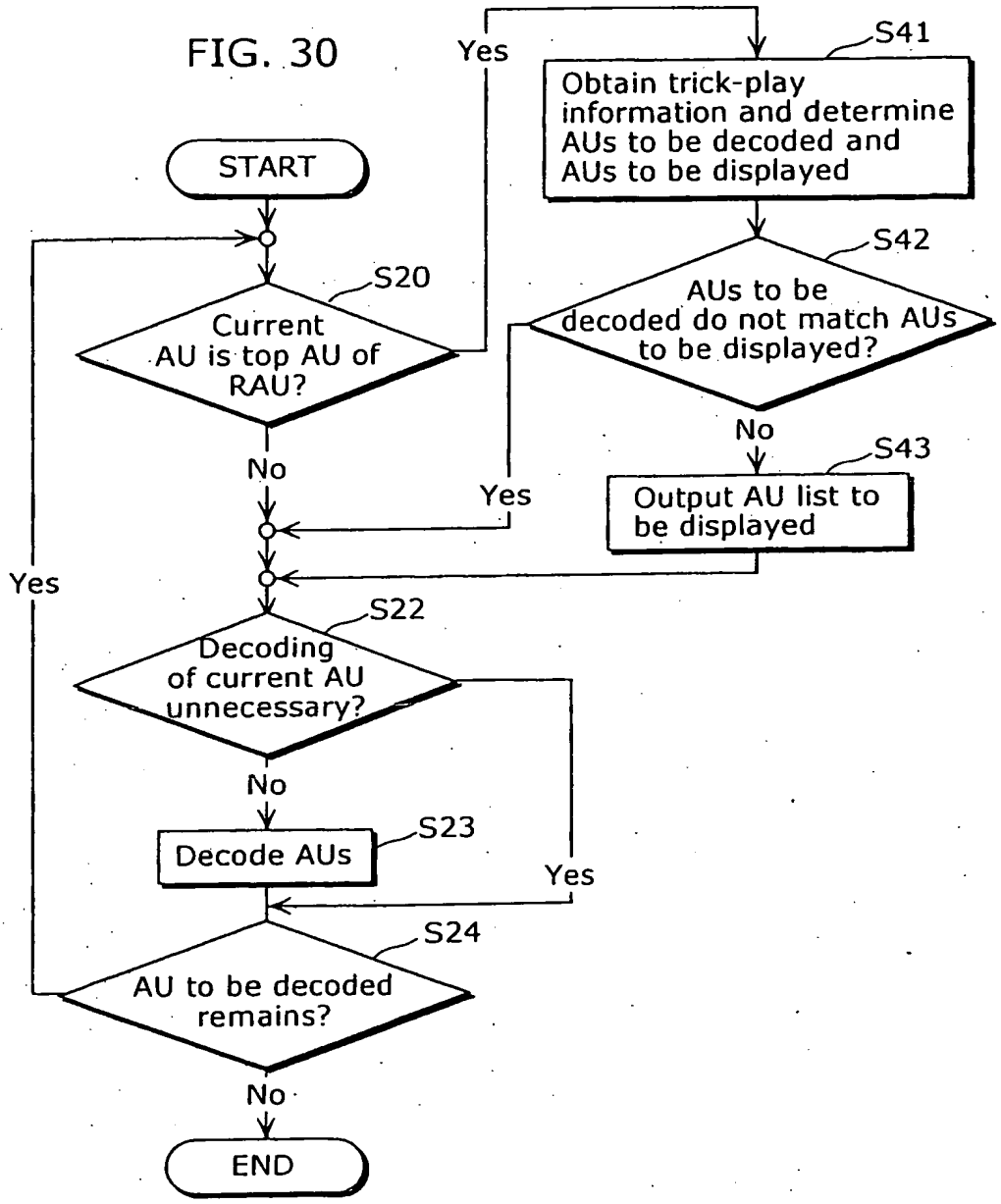

START
S20
Current AU is top AU of RAU?
Yes
No
S41
Obtain trick-play information and determine AUs to be decoded and AUs to be displayed
S42
AUs to be decoded do not match AUs to be displayed?
No
Yes
S43
Output AU list to be displayed
S22
Decoding of current AU unnecessary?
Yes
No
S23
Decode AUs
S24
AU to be decoded remains?
Yes
No
END

FIG. 31

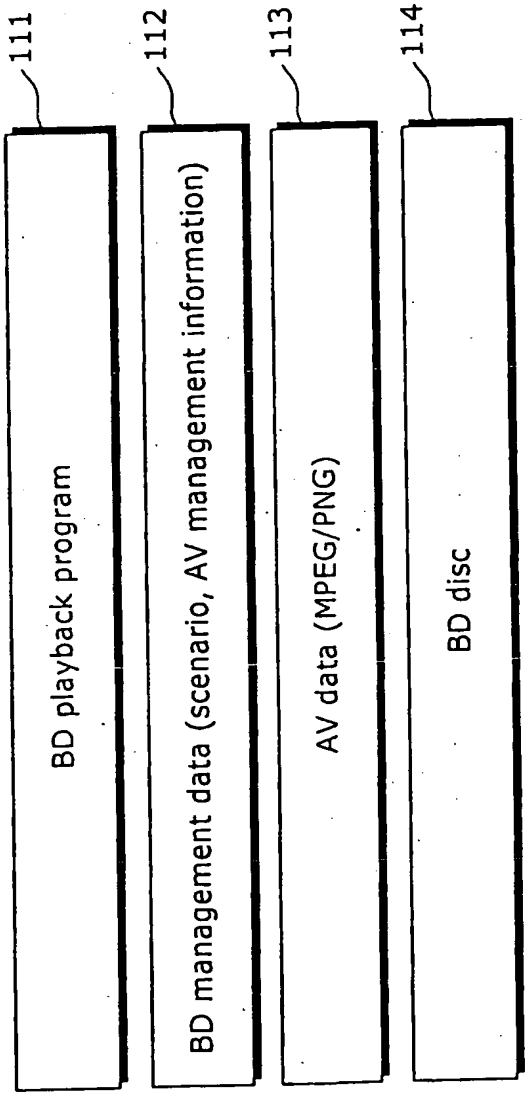

111
112
113
114
BD playback program
BD management data (scenario, AV management information)
AV data (MPEG/PNG)
BD disc

FIG. 32

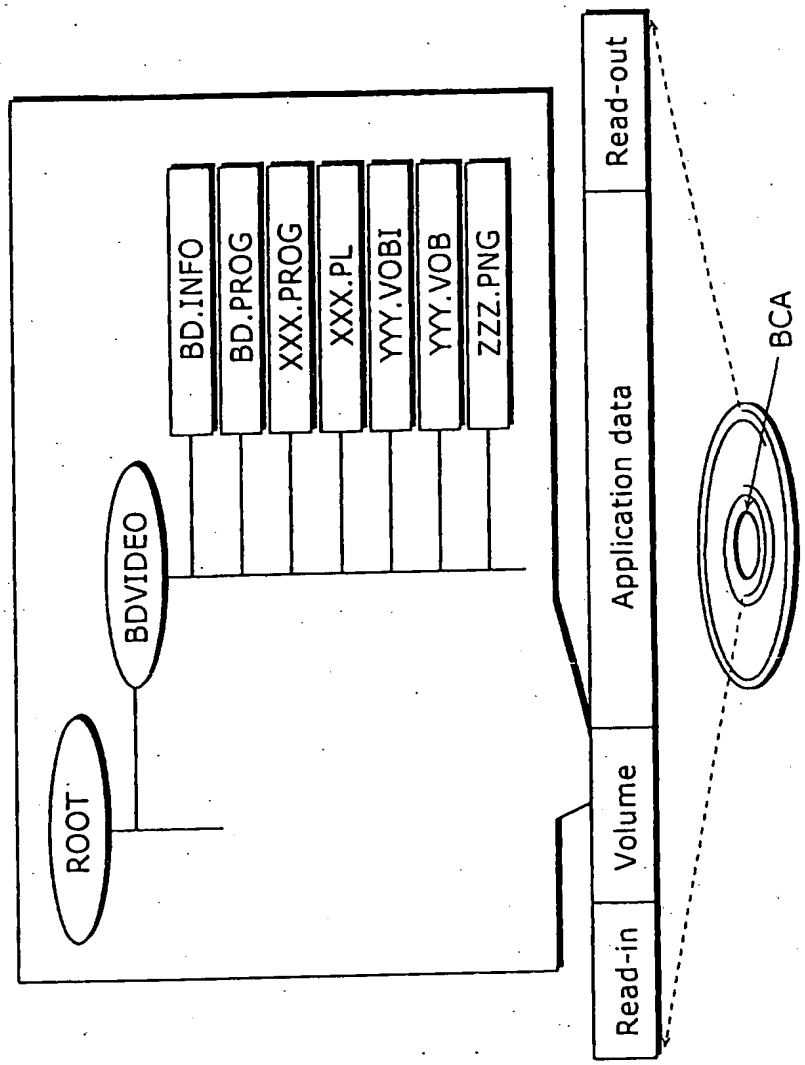

ROOT
BDVIDEO
BD.INFO
BD.PROG
XXX.PROG
XXX.PL
YYY.VOBI
YYY.VOB
ZZZ.PNG
Read-in
Volume
Application data
Read-out
BCA

FIG. 33

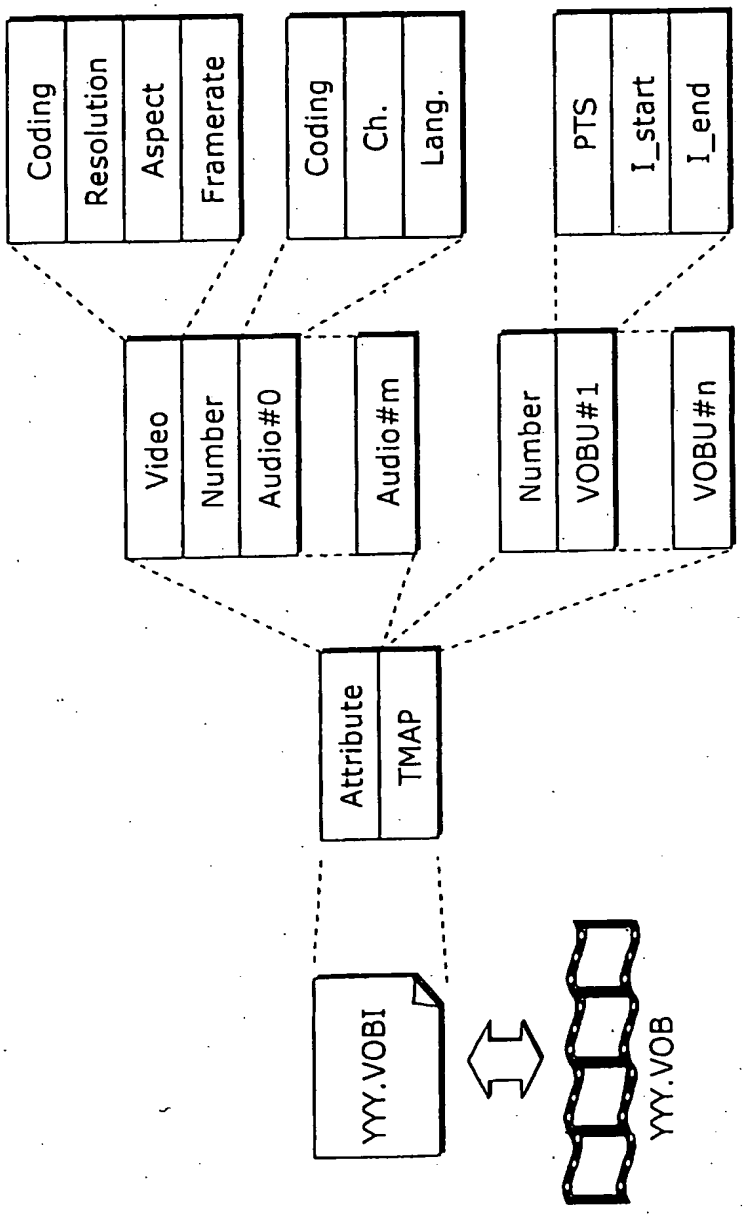
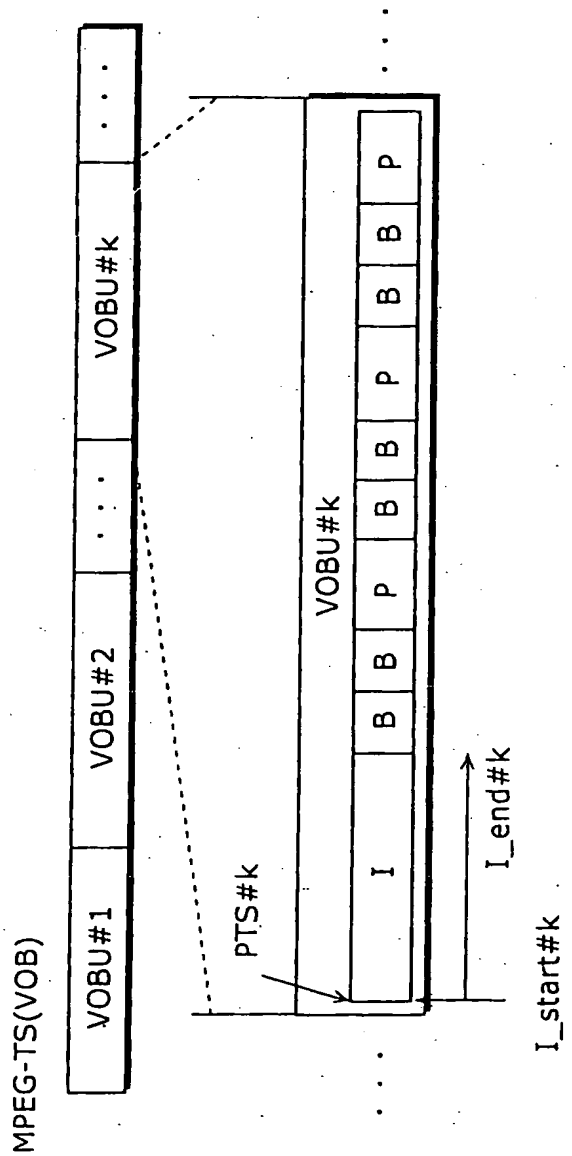

Coding
Resolution
Aspect
Framerate
Coding
Ch.
Lang.
PTS
I_start
I_end
Video
Number
Audio#0
Audio#m
Number
VOBU#1
VOBU#n
Attribute
TMAP
YYY.VOBI
YYY.VOB

FIG. 34


MPEG-TS(VOB)
VOBU#1
VOBU#2
VOBU#k
PTS#k
VOBU#k
I
B
B
P
B
B
P
B
B
P
I_end#k
I_start#k

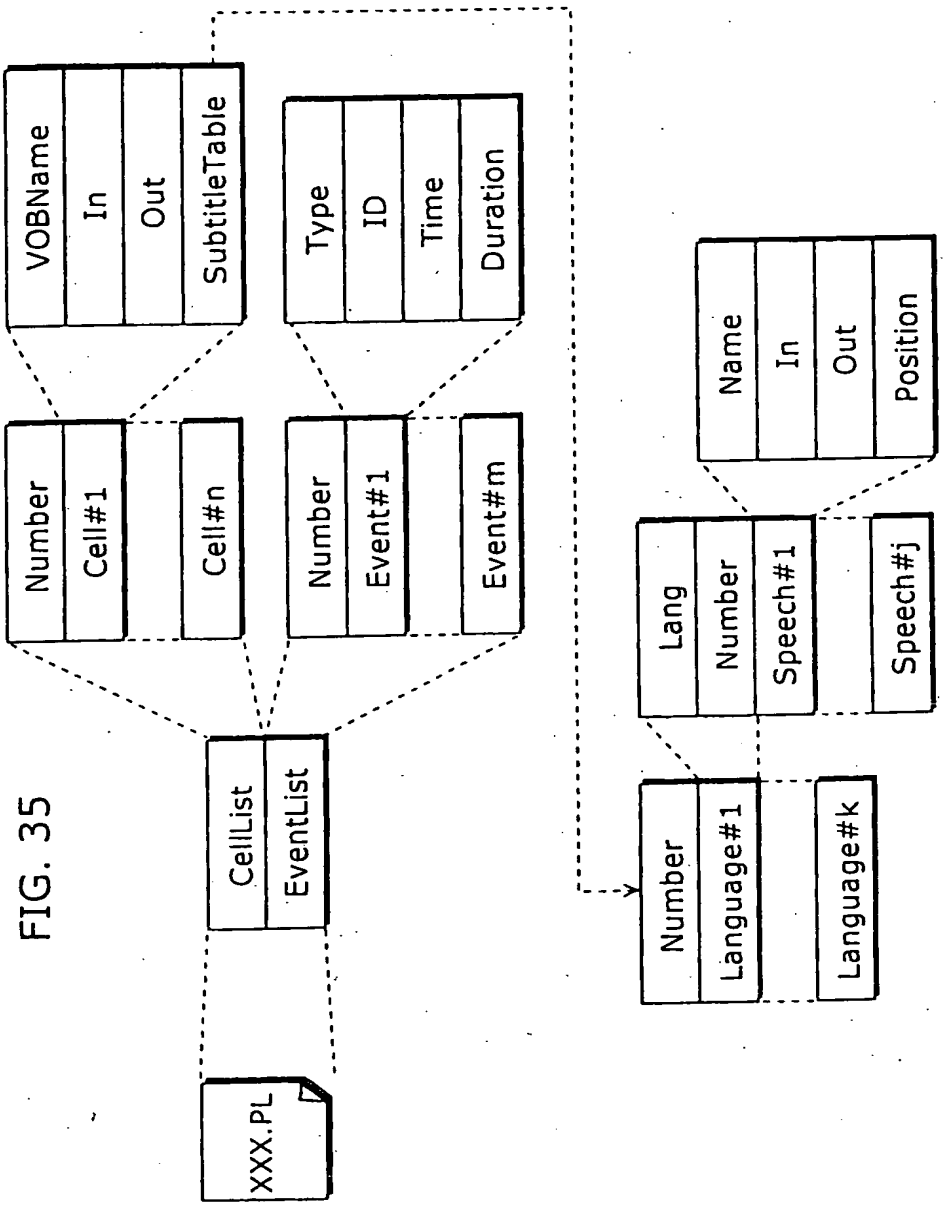

XXX.PL
CellList
EventList
Number
Cell#1
Cell#n
Number
Event#1
Event#m
VOBName
In
Out
SubtitleTable
Type
ID
Time
Duration
Number
Language#1
Language#k
Lang
Number
Speech#1
Speech#j
Name
In
Out
Position

FIG. 35

FIG. 36

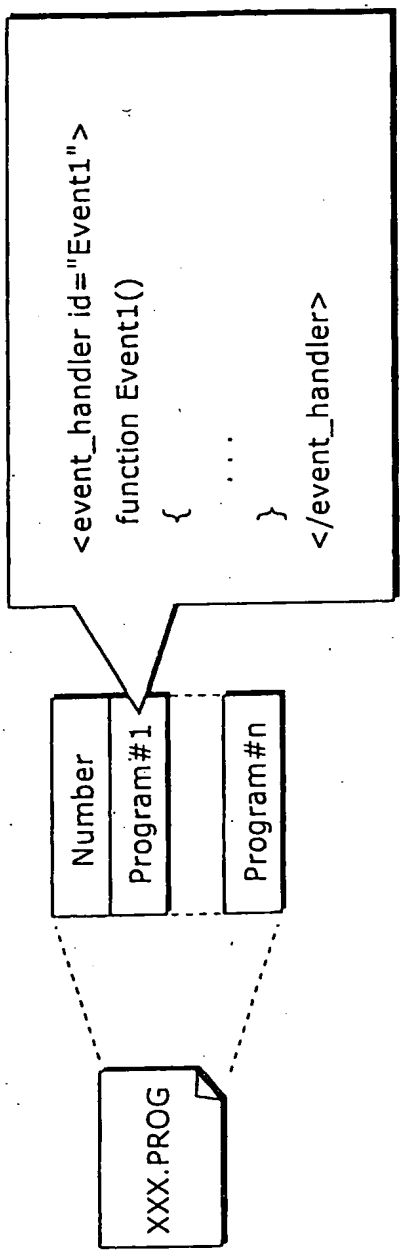

<event_handler id="Event1">
function Event1()
{
...
}
</event_handler>
Number
Program#1
Program#n
XXX.PROG

FIG. 37

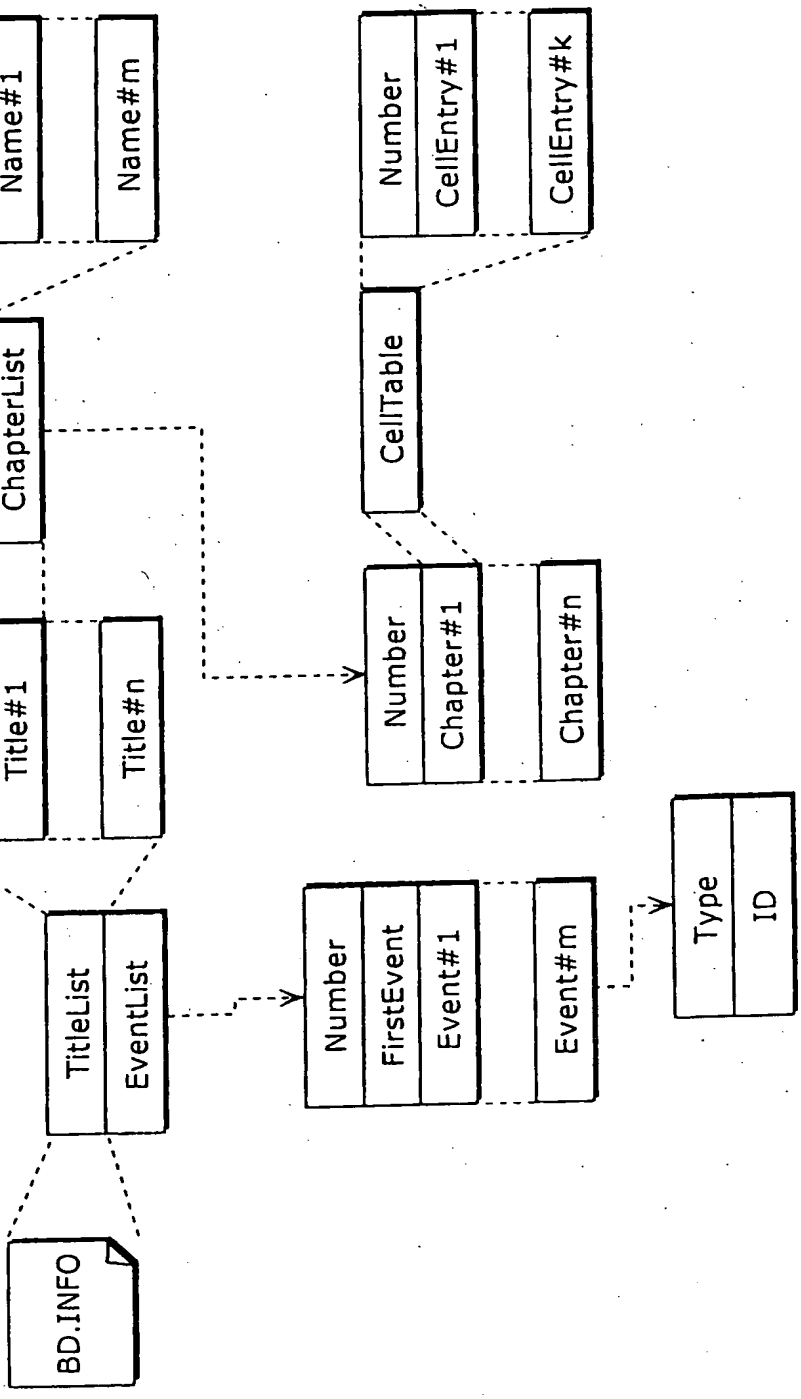

BD.INFO
TitleList
EventList
Number
Title#1
Title#n
PLTable
ChapterList
Number
Name#1
Name#m
Number
FirstEvent
Event#1
Event#m
Type
ID
CellTable
Number
Chapter#1
Chapter#n
Number
CellEntry#1
CellEntry#k

FIG. 38

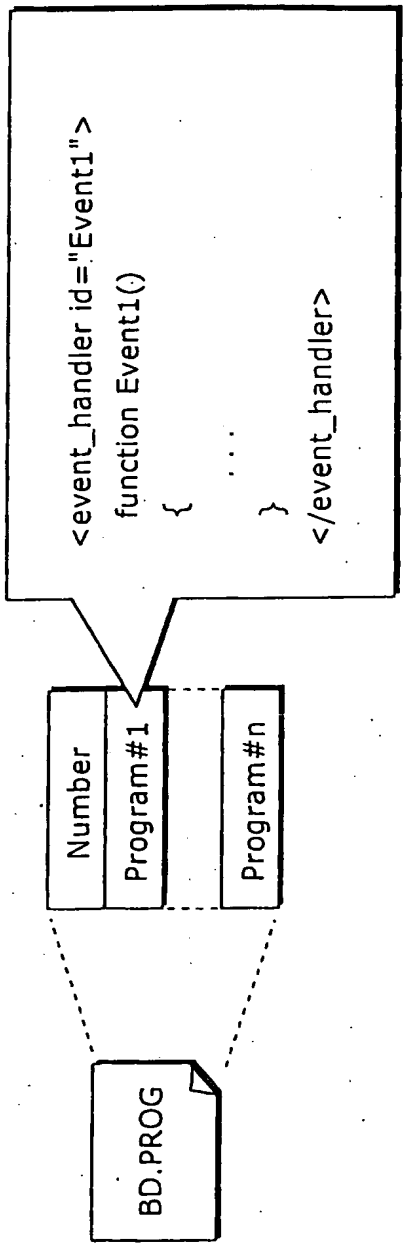

<event_handler id="Event1">
function Event1()
{
...
}
</event_handler>
Number
Program#1
Program#n
BD.PROG

FIG. 39

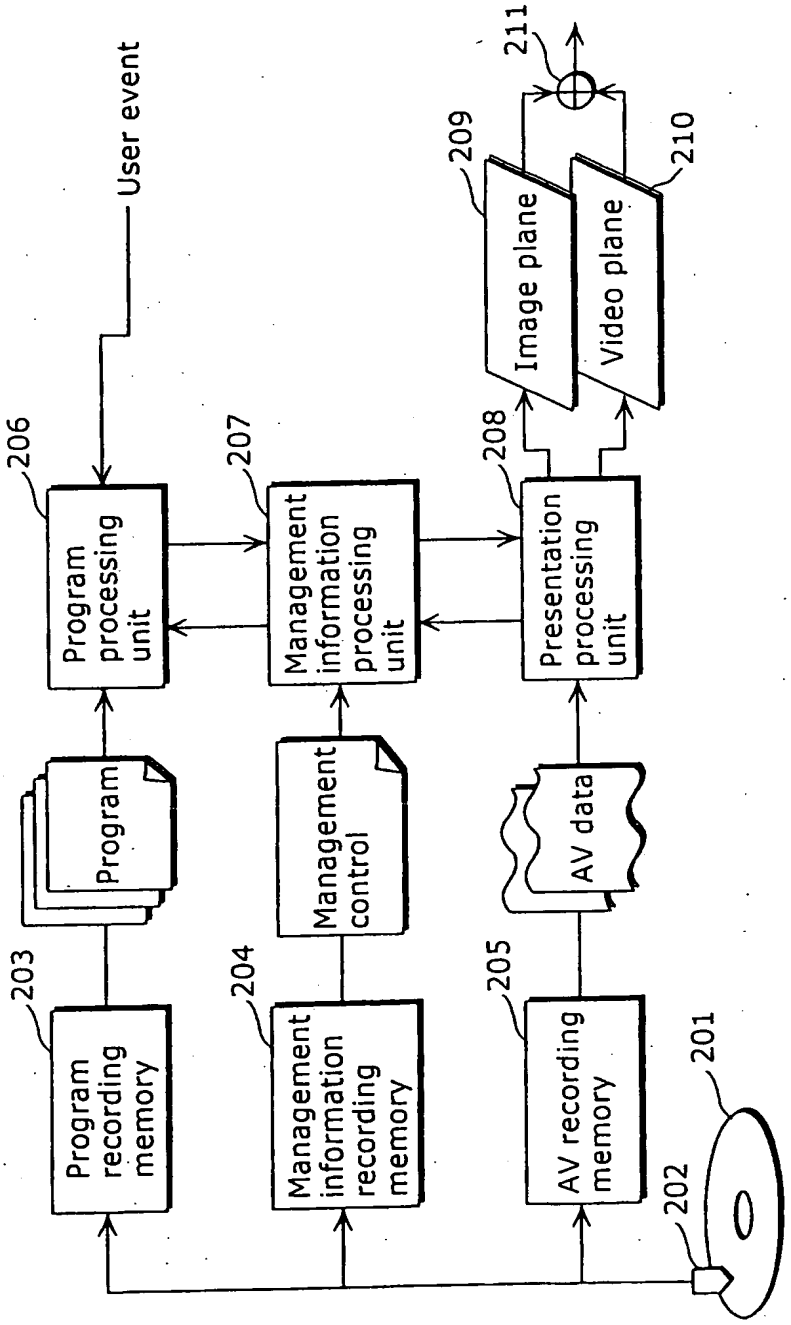

User event
206
Program processing unit
207
Management information processing unit
208
Presentation processing unit
209
Image plane
210
Video plane
211
Program
Management control
AV data
203
Program recording memory
204
Management information recording memory
205
AV recording memory
201
202

Fig. 40A

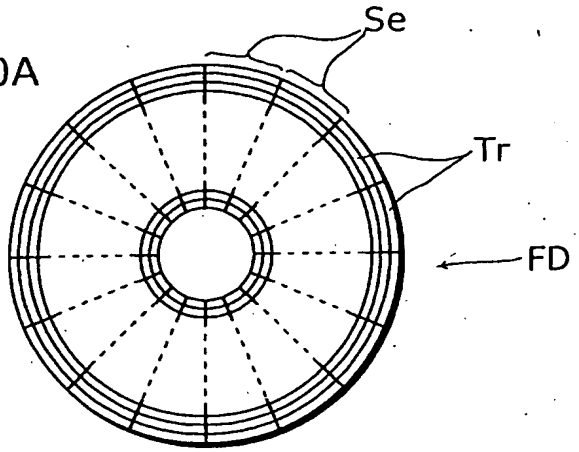


Fig. 40B

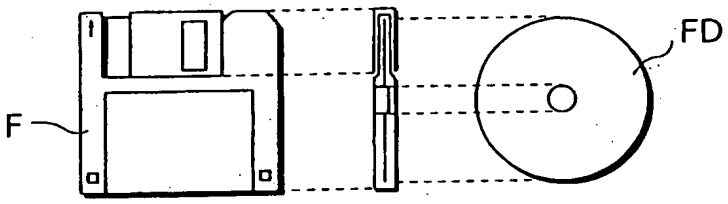


Fig. 40C